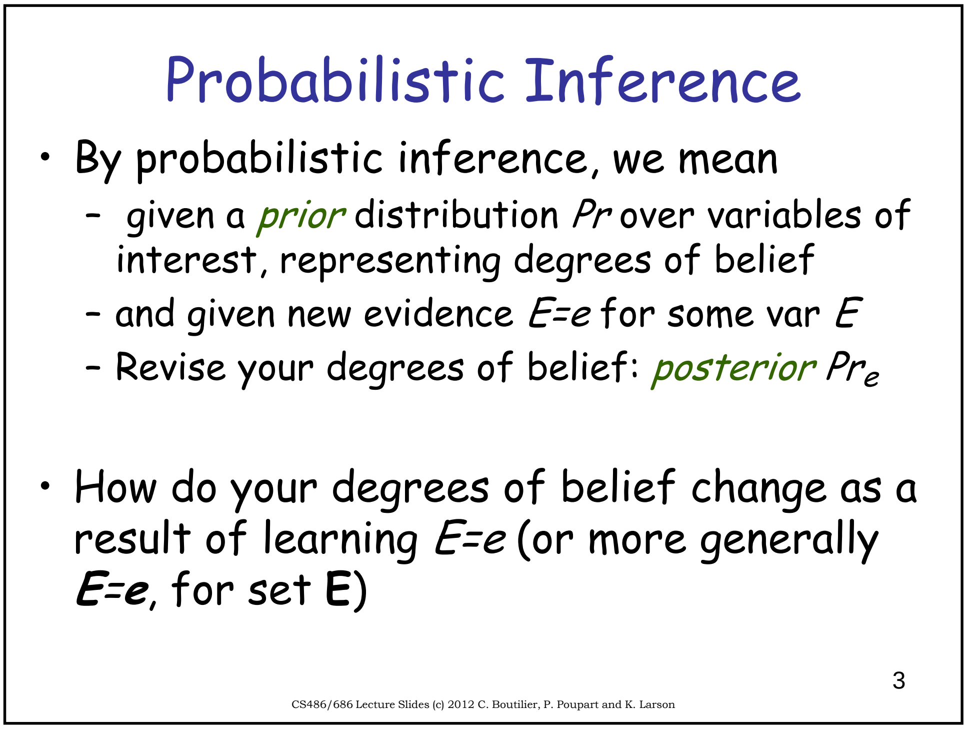
Probabilistic Reasoning
[RN2] Sections 14.1, 14.2
[RN3] Sections 14.1, 14.2

University of Waterloo
CS 486/686
Lecture 8: Oct 4, 2012

Outline

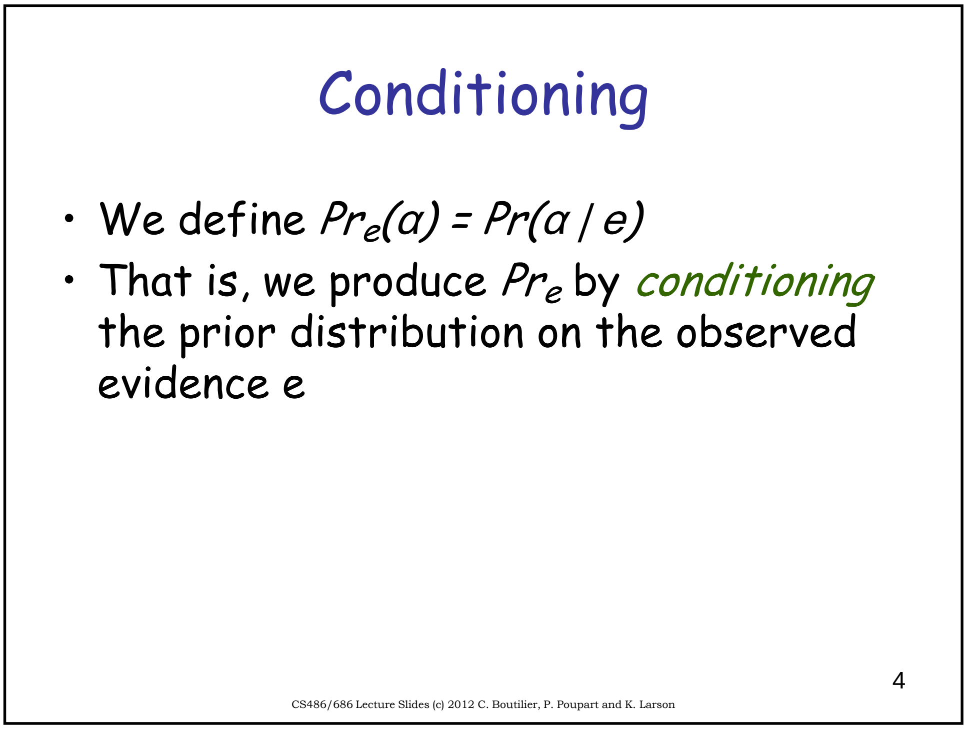
- Review probabilistic inference, independence and conditional independence
- Bayesian networks
  - What are they
  - What do they mean
  - How do we create them



Probabilistic Inference

- By probabilistic inference, we mean
  - given a *prior* distribution Pr over variables of interest, representing degrees of belief
  - and given new evidence $E=e$ for some var E
  - Revise your degrees of belief: *posterior* Pr_e

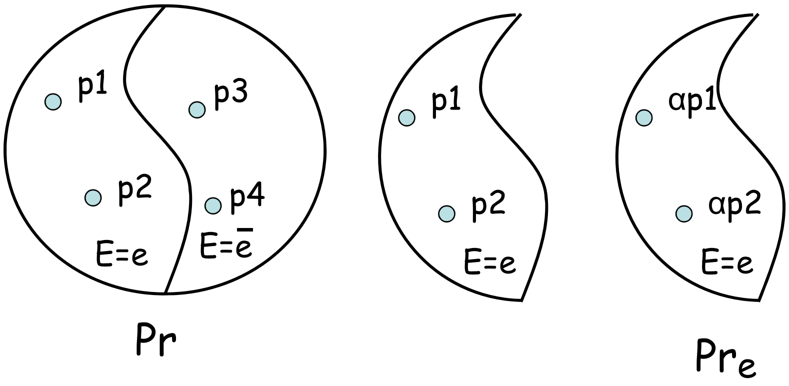
- How do your degrees of belief change as a result of learning $E=e$ (or more generally $\boldsymbol{E}=\boldsymbol{e}$, for set $\mathbf{E}$)



Conditioning

- We define $Pr_e(\alpha) = Pr(\alpha \mid e)$
- That is, we produce Pr_e by *conditioning* the prior distribution on the observed evidence e

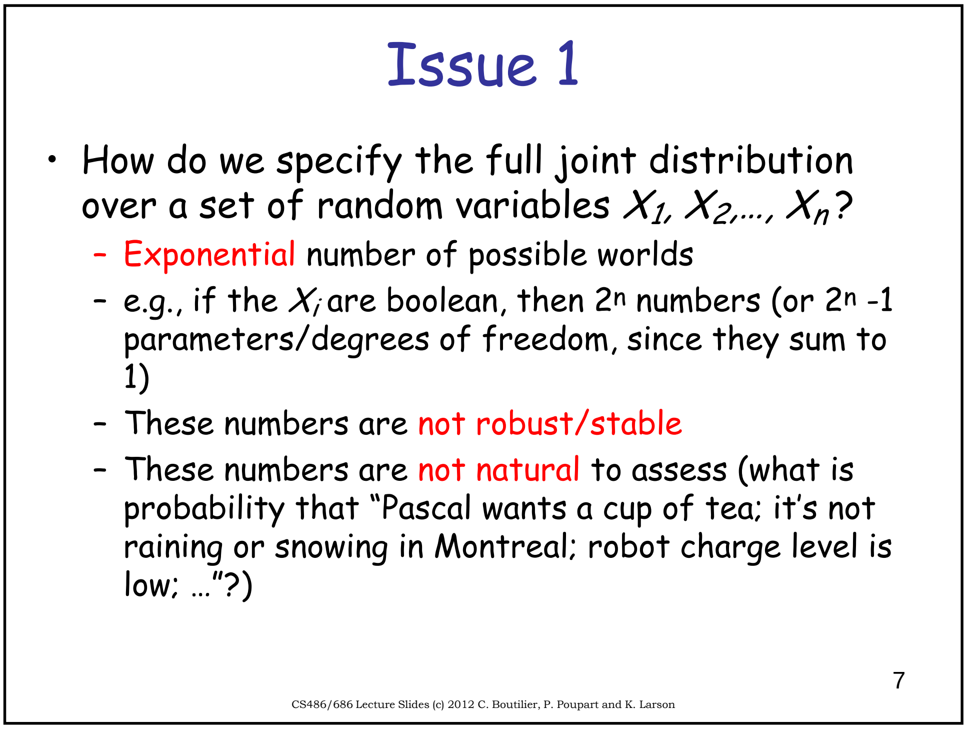
Semantics of Conditioning



α = 1/(p1+p2)
normalizing constant

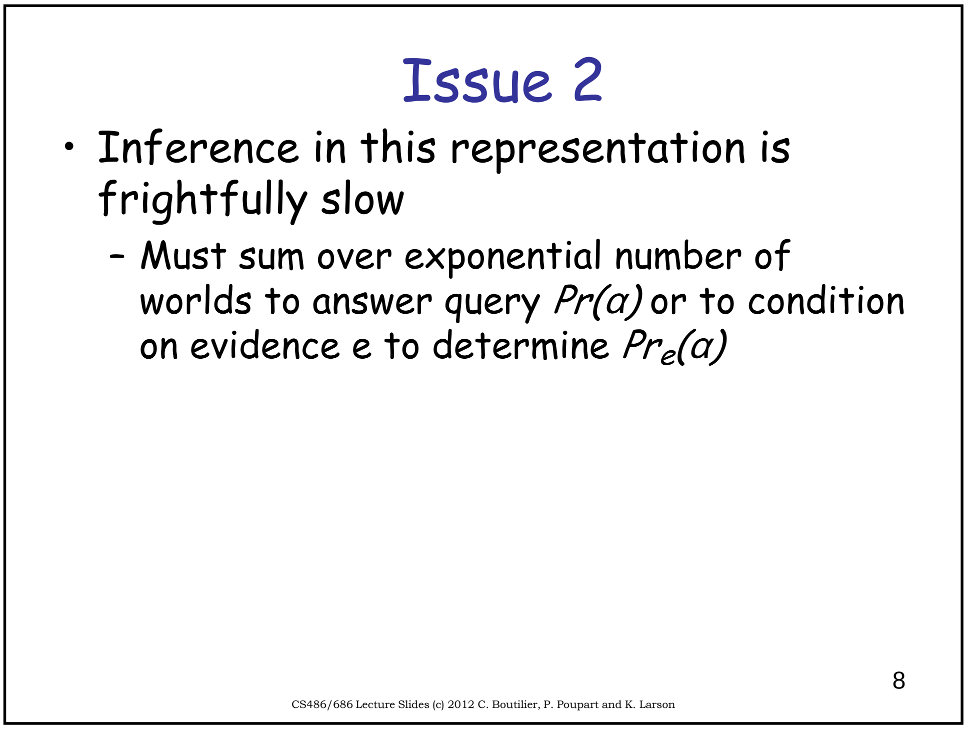
Inference: Computational Bottleneck

- Semantically/conceptually, picture is clear; but several issues must be addressed



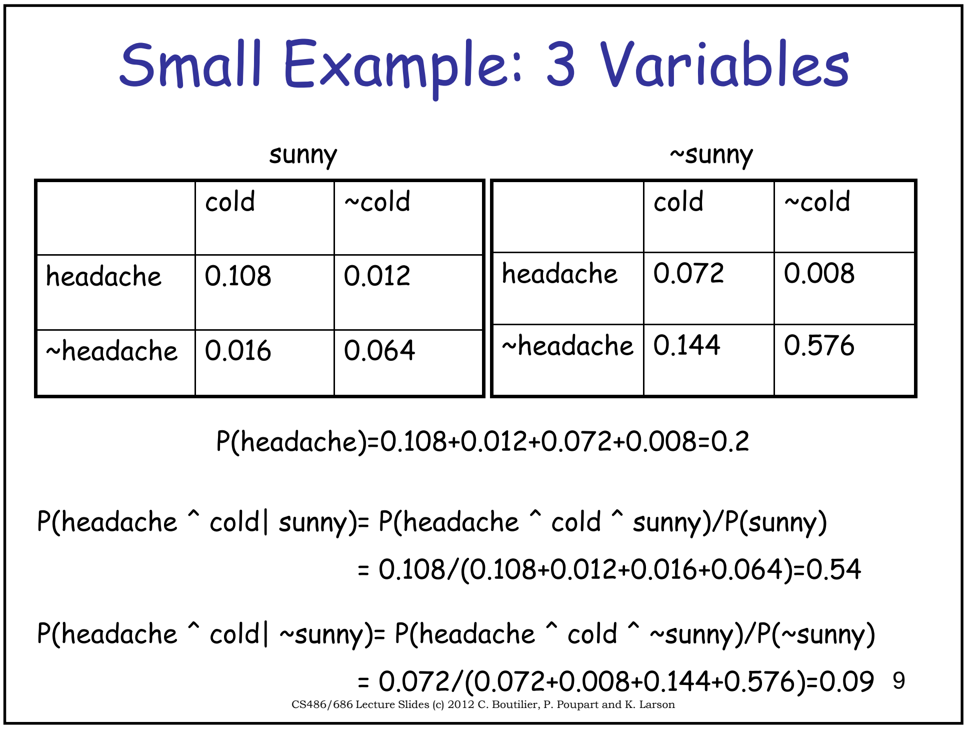
Issue 1

- How do we specify the full joint distribution over a set of random variables $X_1, X_2, \dots, X_n$?
  - Exponential number of possible worlds
  - e.g., if the X_i are boolean, then 2^n numbers (or 2^n -1 parameters/degrees of freedom, since they sum to 1)
  - These numbers are not robust/stable
  - These numbers are not natural to assess (what is probability that "Pascal wants a cup of tea; it's not raining or snowing in Montreal; robot charge level is low; …"?)



Issue 2

- Inference in this representation is frightfully slow
  - Must sum over exponential number of worlds to answer query $Pr(\alpha)$ or to condition on evidence e to determine $Pr_e(\alpha)$



Small Example: 3 Variables

sunny

	cold	~cold
headache	0.108	0.012
~headache	0.016	0.064

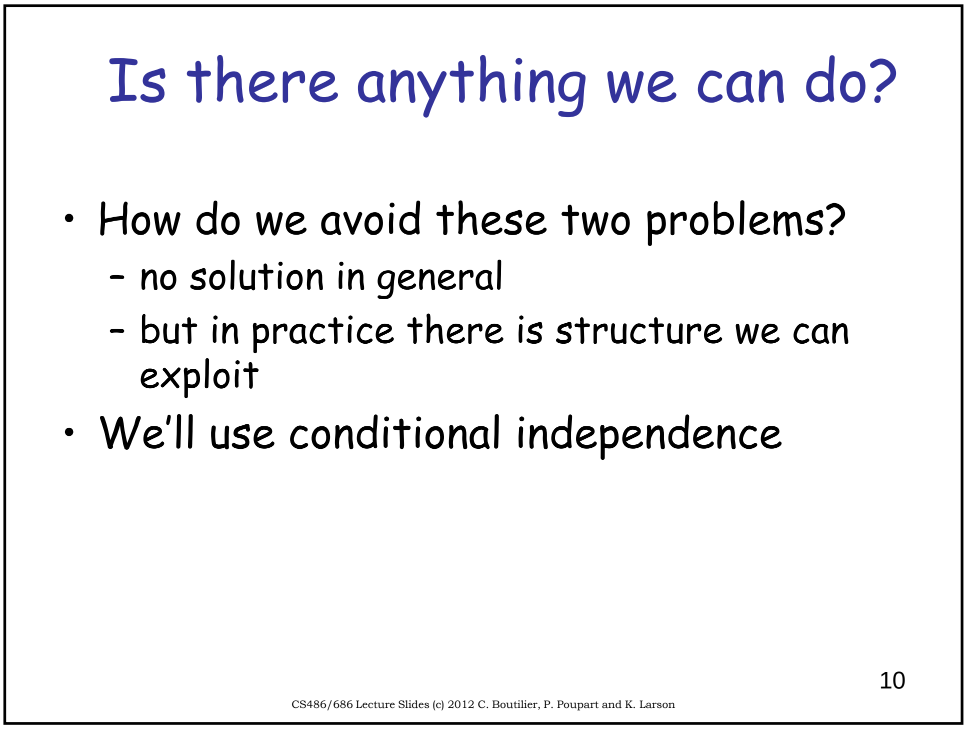
~sunny

	cold	~cold
headache	0.072	0.008
~headache	0.144	0.576

P(headache)=0.108+0.012+0.072+0.008=0.2

P(headache ^ cold| sunny)= P(headache ^ cold ^ sunny)/P(sunny)
= 0.108/(0.108+0.012+0.016+0.064)=0.54

P(headache ^ cold| ~sunny)= P(headache ^ cold ^ ~sunny)/P(~sunny)
= 0.072/(0.072+0.008+0.144+0.576)=0.09



Is there anything we can do?

- How do we avoid these two problems?
  - no solution in general
  - but in practice there is structure we can exploit
- We'll use conditional independence

Independence

- Recall that x and y are *independent* iff:
  - $\Pr(x) = \Pr(x|y)$ iff $\Pr(y) = \Pr(y|x)$ iff $\Pr(xy) = \Pr(x)\Pr(y)$
  - intuitively, learning y doesn't influence beliefs about x
- x and y are *conditionally independent given* z iff:
  - $\Pr(x|z) = \Pr(x|yz)$ iff $\Pr(y|z) = \Pr(y|xz)$ iff
    $\Pr(xy|z) = \Pr(x|z)\Pr(y|z)$ iff …
  - intuitively, learning y doesn't influence your beliefs about x *if you already know z*
  - e.g., learning someone's mark on 486 exam can influence the probability you assign to a specific GPA; but if you already knew the **final** 486 grade, learning the exam mark would *not* influence your GPA assessment

Variable Independence

- Two *variables* X and Y are conditionally independent given variable Z iff x, y are conditionally independent given z for all $x \in Dom(X)$, $y \in Dom(Y)$, $z \in Dom(Z)$
  - Also applies to sets of variables **X**, **Y**, **Z**
  - Also to unconditional case (X,Y independent)
- If you know the value of Z (*whatever* it is), nothing you learn about Y will influence your beliefs about X
  - these definitions differ from earlier ones (which talk about events, not variables)

What good is independence?

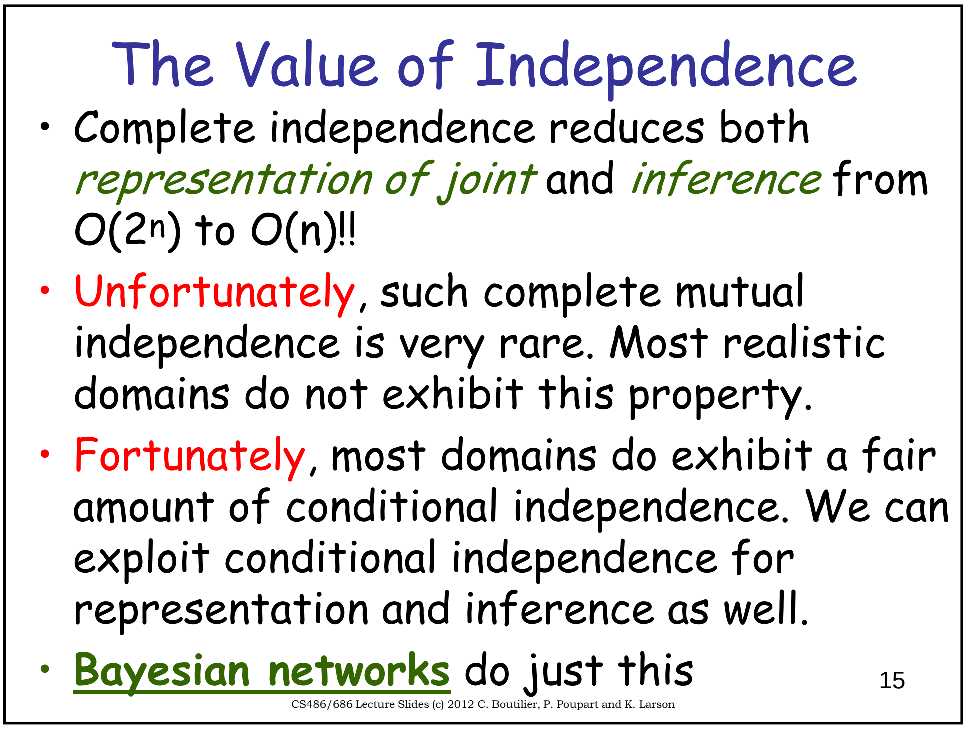
- Suppose (say, boolean) variables $X_1, X_2, \ldots, X_n$ are mutually independent
  - We can specify full joint distribution using only n parameters (linear) instead of $2^n - 1$ (exponential)
- How? Simply specify $Pr(x_1), \ldots Pr(x_n)$
  - From this we can recover the probability of any world or any (conjunctive) query easily
    - Recall P(x,y)=P(x)P(y) and P(x|y)=P(x) and P(y|x)=P(y)

Example

- 4 independent boolean random variables X_1, X_2, X_3, X_4
- $P(x_1)=0.4, P(x_2)=0.2, P(x_3)=0.5, P(x_4)=0.8$

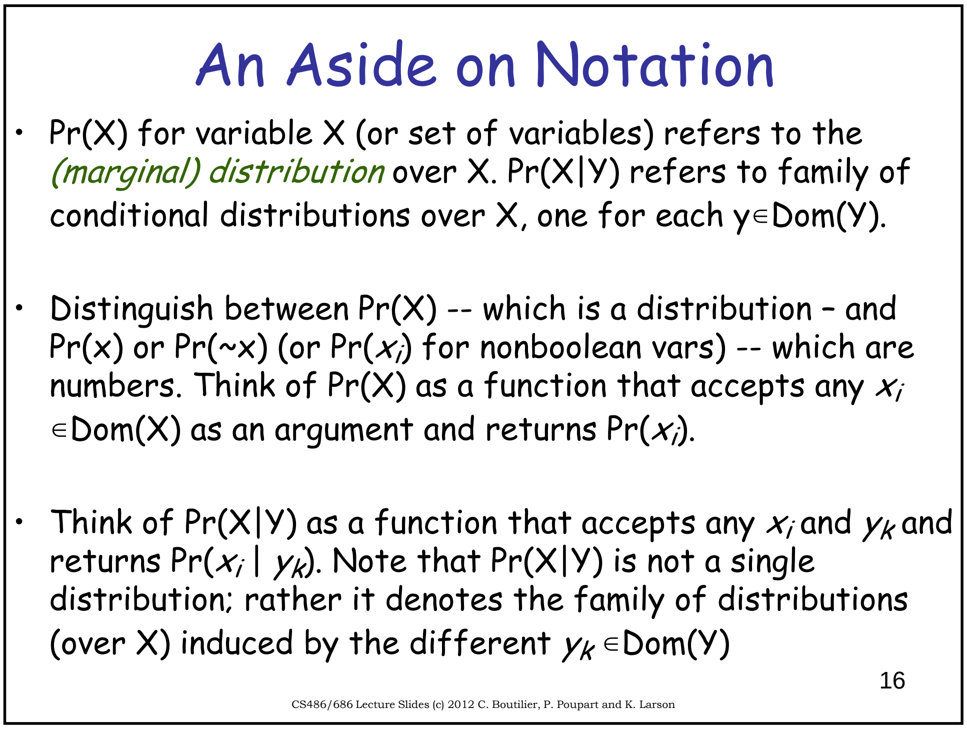
$$P(x1,\sim x2,x3,x4)=P(x1)(1-P(x2))P(x3)P(x4)$$
$$= (0.4)(0.8)(0.5)(0.8)$$
$$= 0.128$$

$$P(x1,x2,x3|x4)=P(x1)P(x2)P(x3)\mathbf{1}$$
$$=(0.4)(0.2)(0.5)(1)$$
$$=0.04$$



The Value of Independence

- Complete independence reduces both *representation of joint* and *inference* from $O(2^n)$ to $O(n)$!!
- Unfortunately, such complete mutual independence is very rare. Most realistic domains do not exhibit this property.
- Fortunately, most domains do exhibit a fair amount of conditional independence. We can exploit conditional independence for representation and inference as well.
- **Bayesian networks** do just this

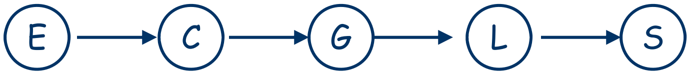


An Aside on Notation

- Pr(X) for variable X (or set of variables) refers to the *(marginal) distribution* over X. Pr(X|Y) refers to family of conditional distributions over X, one for each y∈Dom(Y).
- Distinguish between Pr(X) -- which is a distribution – and Pr(x) or Pr(~x) (or Pr(x_i) for nonboolean vars) -- which are numbers. Think of Pr(X) as a function that accepts any x_i ∈Dom(X) as an argument and returns Pr(x_i).
- Think of Pr(X|Y) as a function that accepts any x_i and y_k and returns Pr(x_i | y_k). Note that Pr(X|Y) is not a single distribution; rather it denotes the family of distributions (over X) induced by the different y_k ∈Dom(Y)

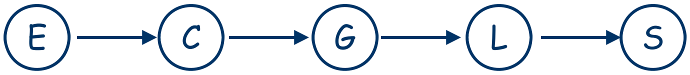
Exploiting Conditional Independence

- Consider a story:
  - If Pascal woke up too early E, Pascal probably needs coffee C; if Pascal needs coffee, he's likely grumpy G. If he is grumpy then it's possible that the lecture won't go smoothly L. If the lecture does not go smoothly then the students will likely be sad S.



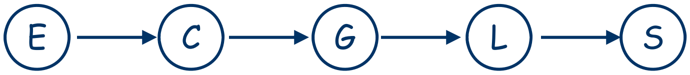
E – Pascal woke up too early G – Pascal is grumpy S – Students are sad
C – Pascal needs coffee L– The lecture did not go smoothly

Conditional Independence



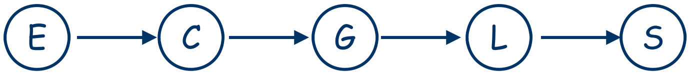
- If you learned any of E, C, G, or L, would your assessment of Pr(S) change?
  - If any of these are seen to be true, you would increase Pr(s) and decrease Pr(~s).
  - So S is *not independent* of E, or C, or G, or L.
- If you knew the value of L (true or false), would learning the value of E, C, or G influence Pr(S)?
  - Influence that these factors have on S is mediated by their influence on L.
  - Students aren't sad because Pascal was grumpy, they are sad because of the lecture.
  - So S is *independent* of E, C, and G, *given* L

Conditional Independence



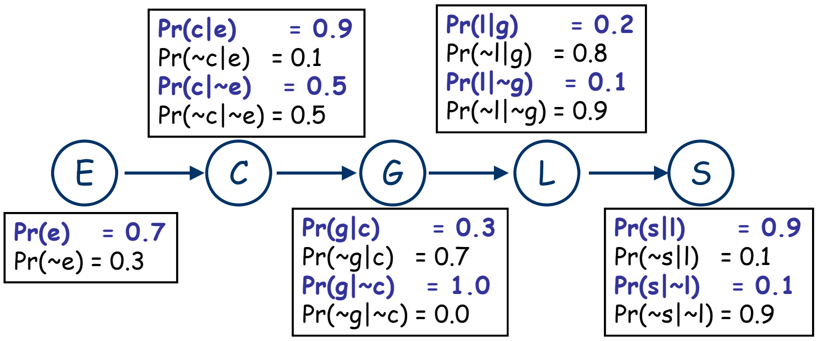
- So S is *independent* of E, and C, and G, *given* L
- Similarly:
  - S is *independent* of E, and C, *given* G
  - G is *independent* of E, *given* C
- This means that:
  - Pr(S | L, {G,C,E}) = Pr(S|L)
  - Pr(L | G, {C,E}) = Pr(L| G)
  - Pr(G| C, {E}) = Pr(G | C)
  - Pr(C | E) and Pr(E) don't "simplify"

Conditional Independence



- By the chain rule (for any instantiation of S…E):
  - Pr(S,L,G,C,E) =
    Pr(S|L,G,C,E) Pr(L|G,C,E) Pr(G|C,E) Pr(C|E) Pr(E)
- By our independence assumptions:
  - Pr(S,L,G,C,E) =
    Pr(S|L) Pr(L|G) Pr(G|C) Pr(C|E) Pr(E)
- We can specify the full joint by specifying five *local conditional distributions*: Pr(S|L); Pr(L|G); Pr(G|C); Pr(C|E); and Pr(E)

Example Quantification



Pr(c|e) = 0.9
Pr(~c|e) = 0.1
Pr(c|~e) = 0.5
Pr(~c|~e) = 0.5

Pr(l|g) = 0.2
Pr(~l|g) = 0.8
Pr(l|~g) = 0.1
Pr(~l|~g) = 0.9

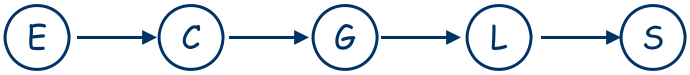
Pr(e) = 0.7
Pr(~e) = 0.3

Pr(g|c) = 0.3
Pr(~g|c) = 0.7
Pr(g|~c) = 1.0
Pr(~g|~c) = 0.0

Pr(s|l) = 0.9
Pr(~s|l) = 0.1
Pr(s|~l) = 0.1
Pr(~s|~l) = 0.9

- Specifying the joint requires only 9 parameters (if we note that half of these are “1 minus” the others), instead of 31 for explicit representation
  - linear in number of vars instead of exponential!
  - linear generally if dependence has a chain structure

Inference is Easy



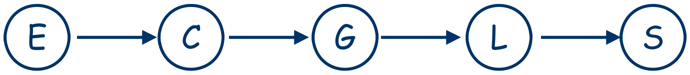
- Want to know P(g)? Use sum out rule:

$$P(g) = \sum_{c_i \in Dom(C)} \Pr(g \mid c_i)\Pr(c_i)$$

$$= \sum_{c_i \in Dom(C)} \Pr(g \mid c_i) \sum_{e_i \in Dom(E)} \Pr(c_i \mid e_i)\Pr(e_i)$$

These are all terms specified in our local distributions!

Inference is Easy



- Computing P(g) in more concrete terms:
  - P(c) = P(c|e)P(e) + P(c|~e)P(~e)
    = 0.8 * 0.7 + 0.5 * 0.3 = 0.78
  - P(~c) = P(~c|e)P(e) + P(~c|~e)P(~e) = 0.22
    - P(~c) = 1 – P(c), as well
  - P(g) = P(g|c)P(c) + P(g|~c)P(~c)
    = 0.7 * 0.78 + 0.0 * 0.22 = 0.546
  - P(~g) = 1 – P(g) = 0.454

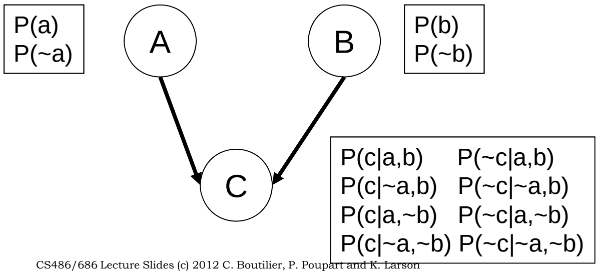
Bayesian Networks

- The structure above is a *Bayesian network*.
  - *Graphical representation* of the direct dependencies over a set of variables + a set of *conditional probability tables (CPTs)* quantifying the strength of those influences.
- Bayes nets generalize the above ideas in very interesting ways, leading to effective means of representation and inference under uncertainty.

Bayesian Networks

aka belief networks, probabilistic networks

- A BN over variables $\{X_1, X_2, \dots, X_n\}$ consists of:
  - a DAG whose nodes are the variables
  - a set of CPTs (Pr(X_i | Parents(X_i)) for each X_i



P(a)
P(~a)

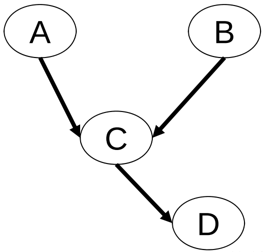
P(b)
P(~b)

P(c|a,b) P(~c|a,b)
P(c|~a,b) P(~c|~a,b)
P(c|a,~b) P(~c|a,~b)
P(c|~a,~b) P(~c|~a,~b)

Bayesian Networks

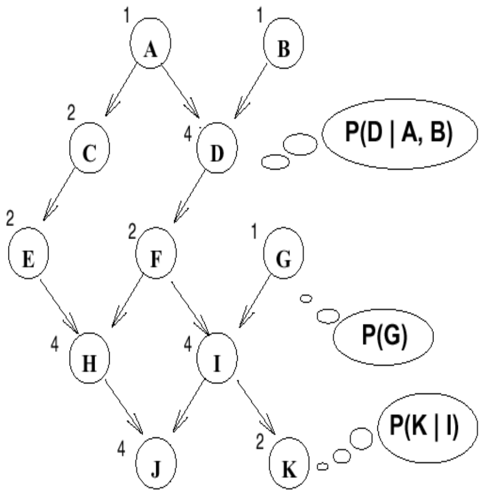
aka belief networks, probabilistic networks

- Key notions
  - parents of a node: Par(X_i)
  - children of node
  - descendents of a node
  - ancestors of a node
  - family: set of nodes consisting of X_i and its parents
    - CPTs are defined over families in the BN



Parents(C)={A,B}
Children(A)={C}
Descendents(B)={C,D}
Ancestors{D}={A,B,C}
Family{C}={C,A,B}

An Example Bayes Net



- A few CPTs are "shown"
- Explicit joint requires 2^{11} -1 =2047 params
- BN requires only 27 parms (the number of entries for each CPT is listed)

Semantics of a Bayes Net

- The structure of the BN means: every X_i is *conditionally independent of all of its nondescendants given its parents*:

$$\Pr(X_i \mid S \cup \mathrm{Par}(X_i)) = \Pr(X_i \mid \mathrm{Par}(X_i))$$

for any subset $S \subseteq \mathrm{NonDescendants}(X_i)$

Semantics of Bayes Nets

- If we ask for $P(x_1, x_2, \ldots, x_n)$ we obtain
  - assuming an ordering consistent with network
- By the chain rule, we have:

  $P(x_1, x_2, \ldots, x_n)$

  $= P(x_n \mid x_{n-1}, \ldots, x_1)\, P(x_{n-1} \mid x_{n-2}, \ldots, x_1) \ldots P(x_1)$

  $= P(x_n \mid Par(x_n))\, P(x_{n-1} \mid Par(x_{n-1})) \ldots P(x_1)$

- Thus, the joint is recoverable using the parameters (CPTs) specified in an arbitrary BN

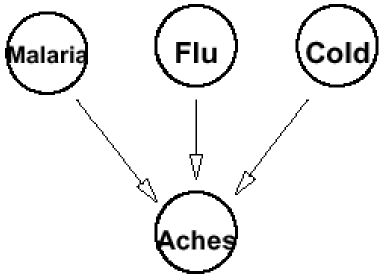
Constructing a Bayes Net

- Given any distribution over variables $X_1, X_2, \ldots, X_n$, we can construct a Bayes net that faithfully represents that distribution.

Take any ordering of the variables (say, the order given), and go through the following procedure for X_n down to X_1. Let Par(X_n) be any subset S $\subseteq \{X_1, \ldots, X_{n-1}\}$ such that X_n is independent of $\{X_1, \ldots, X_{n-1}\}$ - S given S. Such a subset must exist (convince yourself). Then determine the parents of X_{n-1} in the same way, finding a similar S $\subseteq \{X_1, \ldots, X_{n-2}\}$, and so on. In the end, a DAG is produced and the BN semantics must hold by construction.

Causal Intuitions

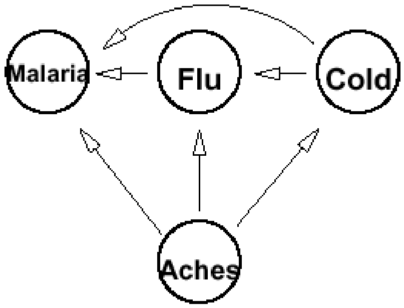
- The construction of a BN is simple
  - works with arbitrary orderings of variable set
  - but some orderings are much better than others!
  - generally, if ordering/dependence structure reflects causal intuitions, a more natural, compact BN results



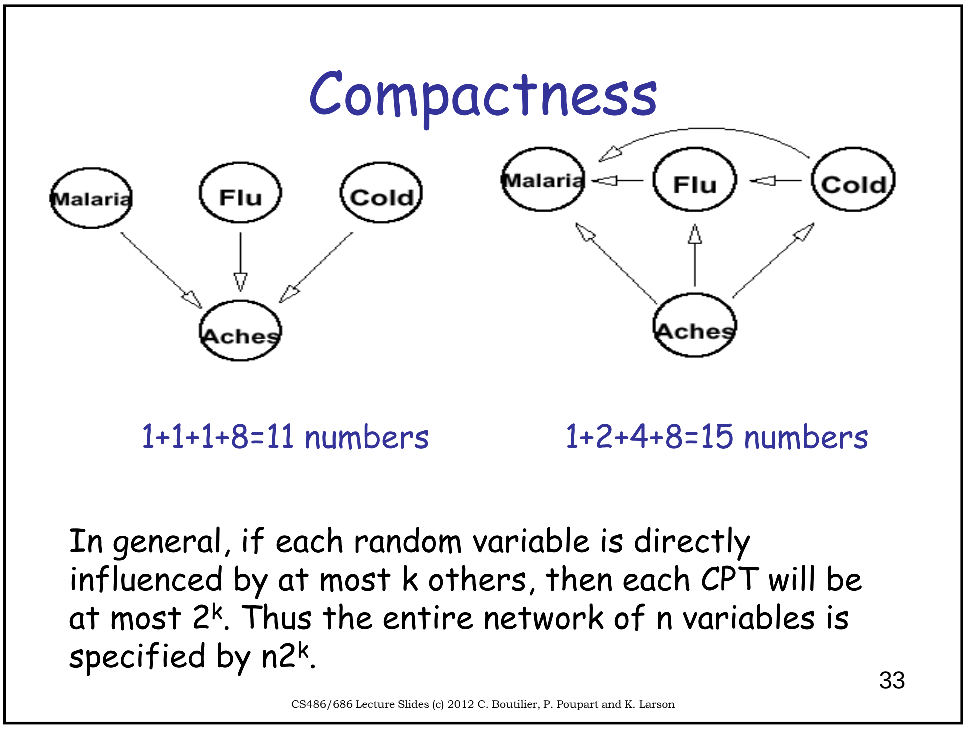
- In this BN, we've used the ordering Mal, Cold, Flu, Aches to build BN for distribution P for Aches
  - Variable can only have parents that come earlier in the ordering

Causal Intuitions

- Suppose we build the BN for distribution P using the opposite ordering
  - i.e., we use ordering Aches, Cold, Flu, Malaria
  - resulting network is more complicated!



- Mal depends on Aches; but it also depends on Cold, Flu *given* Aches
  - Cold, Flu explain away Mal given Aches
- Flu depends on Aches; but also on Cold *given* Aches
- Cold depends on Aches

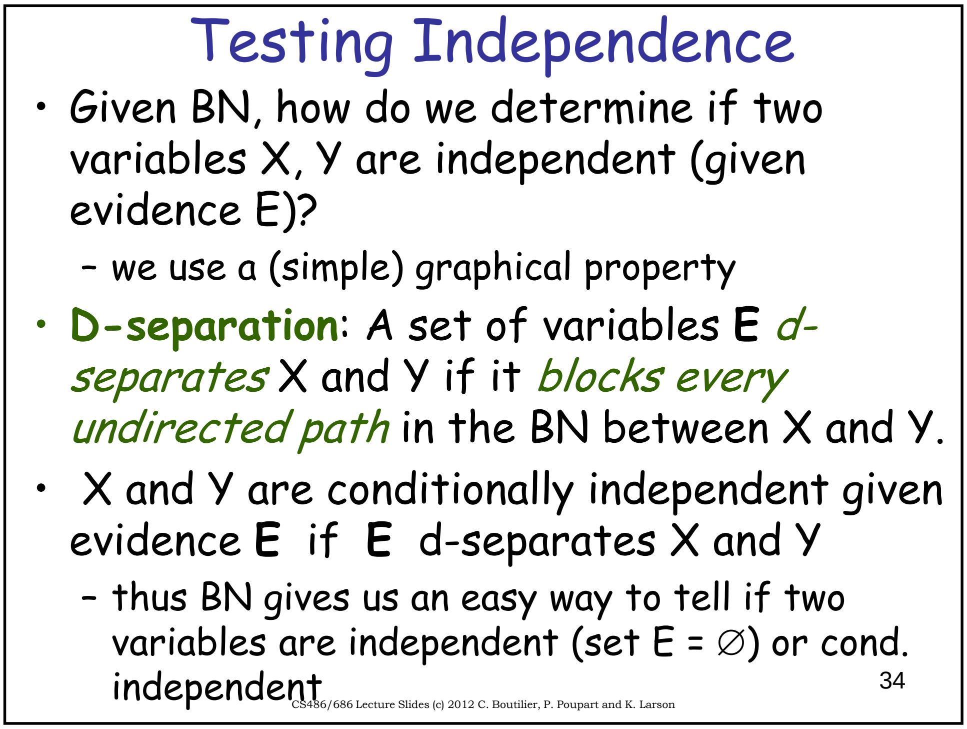


Compactness

1+1+1+8=11 numbers

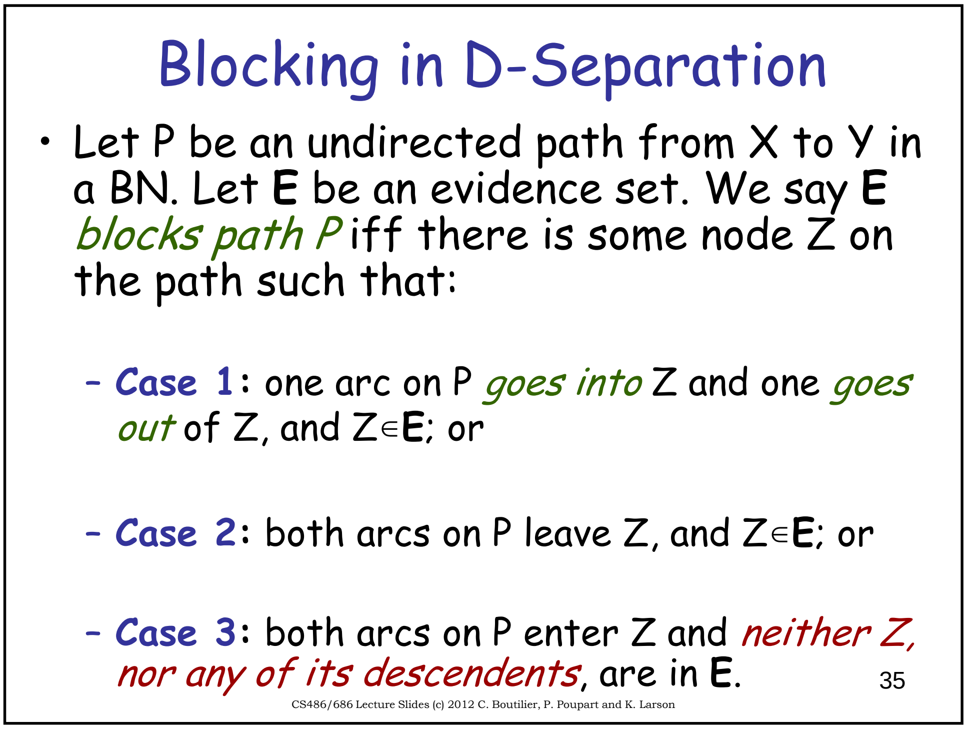
1+2+4+8=15 numbers

In general, if each random variable is directly influenced by at most k others, then each CPT will be at most 2^k. Thus the entire network of n variables is specified by $n2^k$.



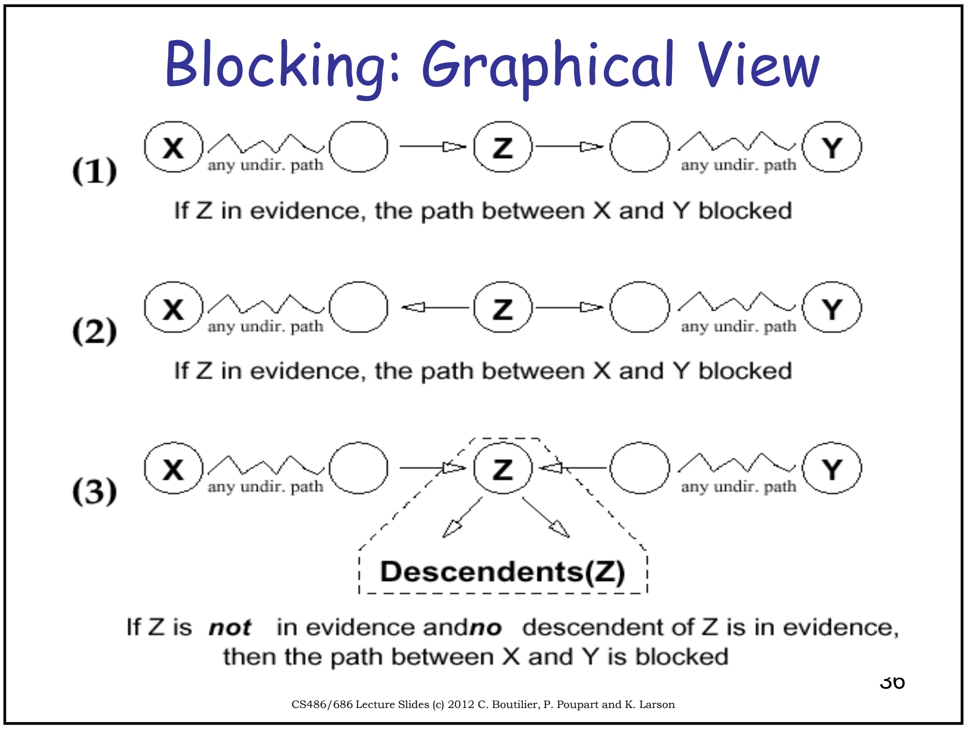
Testing Independence

- Given BN, how do we determine if two variables X, Y are independent (given evidence E)?
  - we use a (simple) graphical property
- **D-separation**: A set of variables **E** *d-separates* X and Y if it *blocks every undirected path* in the BN between X and Y.
- X and Y are conditionally independent given evidence **E** if **E** d-separates X and Y
  - thus BN gives us an easy way to tell if two variables are independent (set E = ∅) or cond. independent



Blocking in D-Separation

- Let P be an undirected path from X to Y in a BN. Let **E** be an evidence set. We say **E** *blocks path P* iff there is some node Z on the path such that:
  - **Case 1**: one arc on P *goes into* Z and one *goes out* of Z, and Z∈**E**; or
  - **Case 2**: both arcs on P leave Z, and Z∈**E**; or
  - **Case 3**: both arcs on P enter Z and *neither Z, nor any of its descendents*, are in **E**.



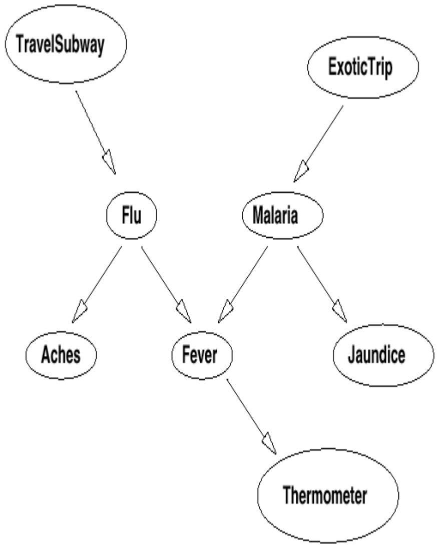
Blocking: Graphical View

If Z in evidence, the path between X and Y blocked

If Z in evidence, the path between X and Y blocked

If Z is ***not*** in evidence and***no*** descendent of Z is in evidence, then the path between X and Y is blocked

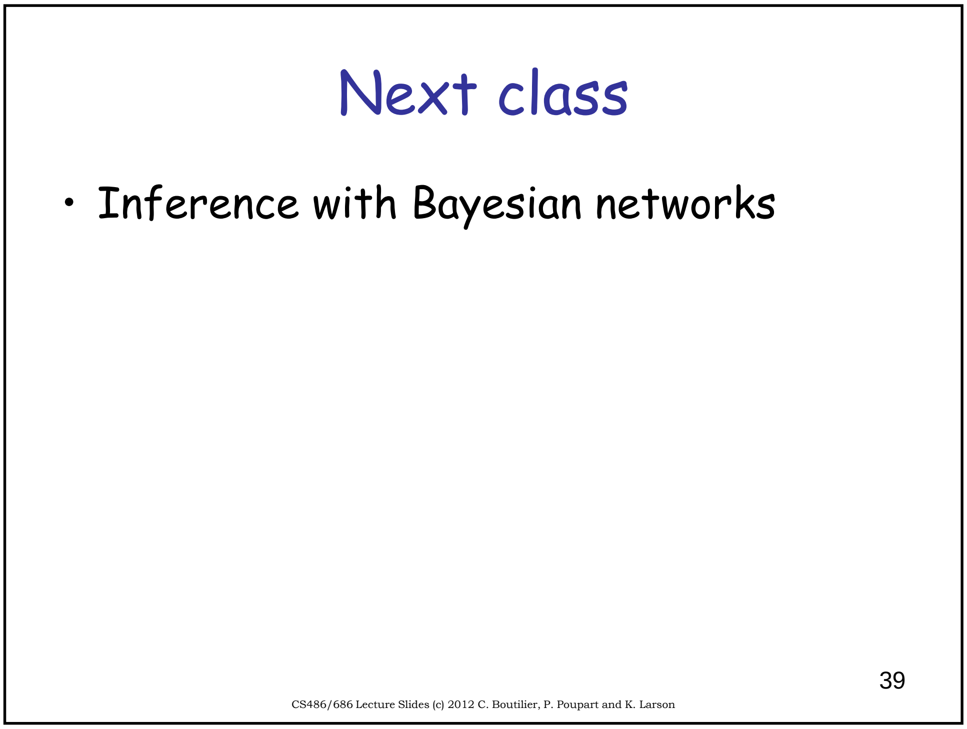
D-Separation: Intuitions



1. Subway and Thermometer?
2. Aches and Fever?
3. Aches and Thermometer?
4. Flu and Malaria?
5. Subway and ExoticTrip?

D-Separation: Intuitions

- Subway and Therm are dependent; but are independent given Flu (since Flu blocks the only path)
- Aches and Fever are dependent; but are independent given Flu (since Flu blocks the only path). Similarly for Aches and Therm (dependent, but indep. given Flu).
- Flu and Mal are indep. (given no evidence): Fever blocks the path, since it is *not in evidence*, nor is its descendant Therm. Flu,Mal are dependent given Fever (or given Therm): nothing blocks path now.
- Subway,ExoticTrip are indep.; they are dependent given Therm; they are indep. given Therm and Malaria. This for exactly the same reasons for Flu/Mal above.



Next class

- Inference with Bayesian networks